CLAIMS

We claim:

1. A method for encoding data, the method comprising:

coding a plurality of events in an event sequence to produce encoded data; and

generating a bitstream using the encoded data, including adding zero or more stuffing bits to the bitstream after the encoded data, wherein the zero or more stuffing bits operate to substantially maintain a relationship between an amount of events encoded, a number of blocks being coded, and a number of bits in the bitstream.

2. The method defined in Claim 1 wherein the one or more stuffing bits comprise one or more stuffing bytes preceded by zero or more alignment bits.

3. The method defined in Claim 2 wherein the one or more stuffing bytes are a pattern recognizable by a decoder.

4. The method defined in Claim 1 wherein an arithmetic coder is used to produce the encoded data from the event sequence.

5. The method defined in Claim 4 wherein the bitstream includes header data.

6. The method defined in Claim 1 wherein the stuffing bits are added to the bitstream after encoded data that includes an encoded end of slice indication.

7. The method defined in Claim 5 further comprising partitioning a picture into the one or more slices, wherein each respective slice includes one or more macro blocks.

8. An arithmetic encoder comprising:

a probability estimator to generate a probability estimate that each event of an event sequence has a particular value, wherein the probability estimator generates the probability estimate in response to corresponding context information for said each event; and

a coding engine coupled to the probability estimator to generate zero or more bits of an information sequence in response to each event and its corresponding probability estimate, wherein the coding engine generates zero or more stuffing bits that are appended to the information sequence after bits generated in response to events, wherein the one or more stuffing bits operate to substantially maintain a relationship between an amount of events encoded, a number of blocks of data being coded, and a number of bits generated by the coding engine in the information sequence.

9. The encoder defined in Claim 8 wherein the one or more stuffing bits comprises one or more stuffing bytes preceded by zero or more alignment bits.

10. The encoder defined in Claim 9 wherein the one or more stuffing bytes are a pattern recognizable by a decoder.

11. The encoder defined in Claim 8 wherein an arithmetic coder is used to produce the encoded data from the event sequence.

12. The encoder defined in Claim 8 wherein the bitstream includes header data.

13. The encoder defined in Claim 8 wherein the coding engine adds the stuffing bits to the information sequence after encoded data containing an encoded end of slice indication.

14. An article of manufacture having one or more recordable media storing instructions thereon which, when executed by a system, cause the system to encode data by:

coding a plurality of events in an event sequence to produce encoded data; and

generating a bitstream using the encoded data, including adding zero or more stuffing bits to the bitstream after the encoded data generated in response to the event sequence, wherein the zero or more stuffing bits operate to substantially maintain a relationship between an amount of events encoded, a number of blocks being coded, and a number of bits in the bitstream.

15. The article of manufacture defined in Claim 14 wherein the one or more stuffing bits comprise one or more stuffing bytes preceded by zero or more alignment bits.

16. The article of manufacture defined in Claim 15 wherein the one or more stuffing bytes are a pattern recognizable by a decoder.

17. . The article of manufacture defined in Claim 14 wherein the stuffing bits are added to the bitstream after encoded data that includes an encoded end of slice indication.

18. An apparatus for encoding data, the apparatus comprising:

means for coding a plurality of events in an event sequence to produce encoded data; and

means for generating a bitstream using the encoded data, including means for adding zero or more stuffing bits to the bitstream after the encoded data generated in response to the event sequence, wherein the one or more stuffing bits operate to substantially maintain a relationship between an amount of events encoded, a number of blocks being coded, and a number of bits in the bitstream.

19. An arithmetic decoder comprising:

a probability estimator to generate a probability estimate that an event of an event sequence has a particular value, wherein the probability estimator generates the probability estimate in response to corresponding context information for said event of the event sequence; and

a decoding engine coupled to the probability estimator to generate an event of an event sequence in response to its corresponding probability estimate and an information sequence, wherein the decoding engine recognizes any stuffing bits in the information sequence and performs no decoding on the zero or more stuffing bits, the zero or more stuffing bits operating to substantially maintain a relationship between an amount of events encoded, a number of blocks being coded, and a number of bits in the information sequence.

20. The arithmetic decoder defined in Claim 19 wherein the stuffing bits comprise one or more stuffing bytes preceded by zero or more alignment bits.

21. The arithmetic decoder defined in Claim 20 wherein the decoding engine recognizes the stuffing bytes by identifying a pattern associated with the stuffing bytes.

22. The arithmetic decoder defined in Claim 19 wherein the decoding engine decodes a header before decoding arithmetically encoded data.

23. The method defined in Claim 19 wherein no renormalization is performed after decoding a last event in the event sequence.

24. A decoding method comprising:

producing a plurality of events of an event sequence from a bitstream comprising encoded data; and

recognizing any stuffing bits following the encoded data in the bitstream without performing decoding on the stuffing bits, the stuffing bits operating to substantially maintain a relationship between an amount of events encoded, a number of blocks being coded, and a number of bits in the bitstream.

25. The method defined in Claim 24 wherein the stuffing bits comprise one or more stuffing bytes preceded by zero or more alignment bits.

26. The method defined in 25 wherein recognizing the stuffing bytes comprises identifying a pattern associated with the stuffing bytes.

27. The method defined in Claim 24 wherein an arithmetic decoder produces the sequence of events from the encoded data.

28. The method defined in Claim 27 further comprising decoding a header before using the arithmetic decoder.

29. The method defined in Claim 27 wherein no renormalization is performed after decoding a last event in the event sequence.

30. An article of manufacture having one or more recordable media storing instructions thereon which, when executed by a system, cause the system to decode data by:

producing a plurality of events of an event sequence from a bitstream comprising encoded data; and

recognizing any stuffing bits following the encoded data in the bitstream without performing decoding on the stuffing bits, the stuffing bits operating to substantially maintain a relationship between an amount of events encoded, a number of blocks being coded, and a number of encoded bits.

31. The method defined in Claim 30 wherein the stuffing bits comprise one or more stuffing bytes preceded by zero or more alignment bits.

32. The method defined in Claim 31 wherein recognizing the stuffing bytes comprises identifying a pattern associated with the stuffing bytes.

33. The method defined in Claim 30 wherein an arithmetic decoder produces the sequence of events from the encoded data.

34. The method defined in Claim 30 further comprising decoding a header before decoding arithmetically encoded data.